

# Performance Prediction using Neural Network and Confidence Intervals: a Gas Turbine application.

Silvia Cisotto, Dr Randa Herzallah

Aston University

Systems Analytics Research Institute (SARI)

Birmingham, UK

cisottos@aston.ac.uk, r.herzallah@aston.ac.uk

**Abstract**— The combination of Condition Based monitoring techniques with the predictive capabilities of neural networks represents a topic of central importance when it comes to maximizing production profits and consequently reducing costs and downtime. The ability to plan the best strategy based on the prediction of potential damaging events can represent a significant contribution, especially for the maintenance function. In fact, optimization of the management of the equipment is a fundamental step to guarantee the competitiveness of companies in the current market. In this paper, a tool based on the implementation of Radial Basis Function Neural Networks was developed to support the maintenance function in the decision-making process. In addition to providing an indication of the status of the equipment, the current approach provides an additional level of information in terms of predicting the confidence interval around the prediction of the neural network. The confidence interval combined with the prediction of the future state of the equipment can be of fundamental importance in order to avoid strategic decisions based on a low level knowledge of the system status or prediction performance of the applied algorithm. The developed tool is tested on the prediction of a naval propulsion system gas turbine performance decay, where the statuses of both the turbine and the compressor of the system are predicted as well as predicting their confidence intervals.

*Radial Basis Function Neural Networks, Industry 4.0, Confidence Interval, Predictive Maintenance.*

## I. INTRODUCTION

In the current market, the ability to react quickly to production problems and minimize downtime and costs has become a fundamental feature for the survival of companies. Furthermore, the variability in terms of both product and volume specifications exacerbates the potential consequences of incorrect handling and maintenance of production equipment [1].

In this context, the maintenance function has increasingly covered a key role in maximizing production performance and minimizing the costs incurred by the companies. Indeed, the cost of the maintenance function has been quantified in the automotive industry to be as high as \$20,000 per minute of downtime [2] whereas, as discussed in [3], maintenance costs represent 15-40 % of total production costs.

A significant innovation in this field is represented by the combination of Condition Based Monitoring (CBM) with the Internet of Things (IoT) concepts and Machine Learning (ML)

principles. This combination has led to what is nowadays called "Industry 4.0" where already known and developed tools, such as CBM, are empowered by the analysis and processing of data collected using IoT and cloud-based solutions and processed using ML applications [4-7].

In this paper, a gas turbine synthetic dataset for marine applications made of several attributes and numerous instances is analyzed. Firstly the attributes are selected in terms of correlation with the system performance decay in order to minimize the complexity. Our analyses and selection of the attributes emphasize the importance of data understanding and pre-processing and show that they may simplify the analysis of the data through dimensionality reduction for example. Motivated by the large number of attributes in the Naval propulsion system and the complexity and high correlation of the hidden patterns that represent fault and healthy conditions of the system, we then implement a Radial Basis Function Neural Network model (RBF) to predict the performance of the system. This is because RBF networks are universal function approximators which can approximate an arbitrary function in a given function class to any degree of accuracy. In addition, RBF models can be trained much faster than other neural networks architectures, such as Multilayer Perceptron Neural Networks for example, and they are very stable, as discussed and demonstrated in [9]. Moreover, they have a less complex architecture with input-hidden-output layers and two layers of weights.

Finally, a metric that quantifies the performance of the applied RBF model for predicting the future behavior of the system given a certain status of the input features is developed and tested. In particular, this metric is based on predicting the accuracy of the neural network prediction by providing a prediction for the residual error as a result of the estimation process. Indeed, the system status information obtained are enriched with the determination of the results confidence interval, determined again using another RBF Neural Network with the same structure as the one used to predict the system performance decay, to support a more reliable and conscious decision-making process in terms of system maintenance.

To reduce the obtained model complexity and simulate real world situation where the sensor noise and uncertainty are unavoidable, we introduced Gaussian noises to the input attributes to add a regularization effect, and to the output to represent measurement noise in the sensors measuring the decays.

Although the separate techniques are already present in the literature, the aim of this paper is to present an approach to the system performance prediction that instead of focusing on the optimization of a single aspect of the analysis, takes into account different aspects, starting from the data pre-processing to the use of neural networks as universal approximators to the use of confidence interval prediction with the scope of supporting the maintenance strategic decision-making process.

This paper is organized in 8 Sections. In Section 2, the theory behind the RBF model is introduced. Section 3 and 4 describe the dataset used in the experiment, as well as its pre-processing and selection of the most relevant features with the aim of dimensionality reduction. In Section 5, the introduction of noise both in the input and the output features of the dataset is discussed. The development of a metric to quantify the accuracy of prediction is presented in Section 6. In Section 7, the results obtained by a set of experiments run on the RBF are shown. Conclusion remarks are given in Section 8.

## II. RADIAL BASIS FUNCTIONS

Radial Basis Function Neural Network (RBF) are a class of neural networks made of an input layer, a hidden layer, and an output layer. The activation function of the hidden layer neurons is specified by the distance between the input vector and the prototype or target vector [8]. The RBF architecture encompasses input-hidden-output layers and two layers of weights. The first layer of weights is dedicated to the parameters of the basis functions while the second layer represents the linear combinations of the basis function activation functions.

The output of the RBF network is calculated as [8]:

$$y_k(x) = \sum_{j=1}^M w_{kj} \cdot \phi_j(x) + w_{k0} \quad (1)$$

Where:

- $x$  is the input vector;
- $j = [1, \dots, M]$  is the  $j$ -th hidden neuron;
- $k = [1, \dots, c]$  is the  $k$ -th output neuron;
- $w_{kj}$  are the weights from the hidden neurons to output  $k$ ;
- $\phi_j(x)$  is the activation function of the hidden neuron  $j$ ;
- $w_{k0}$  is the bias weight of output  $k$ .

There are different types of activation functions, such as the Thin Plate Spline, Gaussian, and the Logarithmic. In this paper the Gaussian activation function has been chosen:

$$\phi_j(x) = \exp\left(\frac{-r_j^2}{2\sigma_j^2}\right) \quad (2)$$

Where:

- $r_j = |x - \mu_j|$  is the distance between the input vector and the vector of the centres of the basis function  $\phi_j$ ;
- $\sigma_j$  is the width of  $j$ -th hidden neuron basis function.

Being a supervised learning algorithm, the RBF network encompasses three phases.

The first phase is the training phase, which is made of two steps: firstly, the radial functions are determined by unsupervised techniques on the input data and then the hidden layer weights are found using fast linear supervised methods. In the first step, the input vector is used to determine the basis function parameters  $\mu_j$  and  $\sigma_j$  for the Gaussian activation function for each hidden neuron. Then, keeping these parameters fixed, the second layer weights can be found by linear matrix inversion techniques. So, if the bias in equation 1 is included in the weights assuming that for the bias, the activation function value,  $\phi_0 = 1$ :

$$y_k(x) = \sum_{j=0}^M w_{kj} \cdot \phi_j(x) \rightarrow y_k(x) = W \cdot \phi \quad (3)$$

At this point, the weights can be optimized by minimizing the error between the prediction and the target calculated as:

$$E = \frac{1}{2} \sum (y - t)^2 \quad (4)$$

Therefore, the error function is a quadratic function of the weights and the weights can be determined as:

$$\phi^T \cdot \phi \cdot W^T = \phi^T \cdot T \quad (5)$$

which can be written also as:

$$W^T = \phi^\dagger \cdot T \quad (6)$$

where  $\phi^\dagger$  is the pseudo-inverse of  $\phi$ .

For further details please refer to [8].

## III. DATA

The analyzed dataset in this paper is an open access synthetic dataset generated from a Simulink® model of a Naval Gas Turbine [10] and it can be found at: (<https://archive.ics.uci.edu/ml/machine-learningdatabases/00316/>).

The simulated naval gas turbine system model is not discussed in this paper, but further information can be found in [10].

The Gas Turbine model is made of 16 input features, listed in Table 1 and two outputs, the Compressor Decay coefficient and the Turbine Decay coefficient. The first output variable is related to the decay of the performance of the gas turbine compressor and it varies in the range [0.95; 1], where 0.95 means that a 5% decay in the compressor performances is

recorded. The second output variable is the system turbine decay and, in this case, it varies in the range [0.975; 1].

TABLE I. INPUT FEATURES OF GAS TURBINE SYSTEM.

Feature ID Number	Description
1	Lever position
2	Ship speed
3	Gas Turbine shaft torque
4	Gas Turbine rate of revolutions
5	Gas Generator rate of revolutions
6	Starboard Propeller Torque
7	Port Propeller Torque
8	HP Turbine exit temperature
9	GT Compressor inlet air temperature
10	GT Compressor outlet air temperature
11	HP Turbine exit pressure
12	GT Compressor inlet air pressure
13	GT Compressor outlet air pressure
14	Gas Turbine exhaust gas pressure
15	Turbine Injection Control
16	Fuel Flow

In total, the dataset is composed of 11,934 samples where: 459 samples are purely related to the Compressor decay with no decay in the turbine, 234 samples are pure turbine decay related, and the remaining samples represent a combination of decay status of both the compressor and the turbine. In this paper, the entire dataset is considered in the experiments with the aim of predicting the decay of system regardless if it is generated by the compressor decay, the turbine decay or both.

As already mentioned in the introduction, RBF neural network is a universal nonlinear function approximator that can make data-driven predictions or decisions and they can be trained much faster than other neural networks architectures as well as being more stable, making them able to handle big data in an efficient way. As such, we use an RBF neural network to provide predictions for the gas turbine performance decay.

Three data sets are commonly used in different stages of the RBF model creation. The RBF model parameters are initially optimized on a training data set. The optimal number of hidden neurons in the RBF model is decided in the validation phase where the optimal number of hidden neurons is selected by calculating the error between the network prediction of the validation data set and their actual values. In this paper, the optimal structure in terms of number of centers or hidden neurons has been determined by calculating the

minimum of the Normalized Error (NE) of the validation phase, where the normalized error is calculated as:

$$NE = \frac{(t - y)^2}{(t - \bar{t})^2} \quad (7)$$

Where:

- $t$  is the target vector;
- $y$  is the predicted output vector,
- $\bar{t}$  is the mean of the target vector.

In the final stage of the RBF model creation, the performance of the specified optimal structure is tested on the test dataset which is a sub-set of the original dataset that has normally never been seen before by the network. In order to give a measure of the reliability of the constructed RBF model, the confidence interval has been implemented in this paper, as discussed in more details in Section 5.

The dataset has been divided by randomly sampling the original dataset without sample replacement into three different sub-sets: 50% of data (5967 samples) are used for the training phase of the RBF network, 30% (3580 samples) are used for the validation phase, and the rest 20% (2387 samples) are used for the test phase.

#### IV. DATA PREPROCESSING

Before going on to the actual analysis, the dataset was pre-processed in order to remove the non-relevant features for the prediction of the system performance decay. The main objective of this step is the dimensionality reduction and the avoidance of the “Garbage-in Garbage-out” effect [11].

This is an important data analysis stage that is very often overlooked. Nevertheless, missing this step lead to an increase in the computational capacity needed to perform the task as well as an increase in the model complexity. This will subsequently introduce unneeded noise, time delays in the results calculations, and more in general, reduction in the algorithm predictions performances.

The preprocessing of the dataset mainly includes the calculation of the correlation coefficients between the 16 input features and the 2 output variables. The calculated correlation coefficients are shown in Table 2.

The Correlation coefficient has been calculated as:

$$\rho_{X,Y} = COV[X,Y]/(\sigma_X \cdot \sigma_Y) \quad (8)$$

where:

- $COV[X,Y] = E[(X - \mu_X) \cdot (Y - \mu_Y)]$  is the covariance;
- $\sigma_X, \sigma_Y$  are the standard deviation of X and Y;
- $E$  is the expectation;
- $\mu_X, \mu_Y$  are the means of X and Y.

This coefficient can vary in the range [-1; 1] where the positive value indicates a direct correlation and the negative indicates an inverse correlation.

Based on the value of the correlation coefficients, feature 1 and 2 (lever position and sheep speed) have been removed due to their low correlation with the outputs and the fact that they are both included in the constitutive model of the Gas Turbine [10]. Features 9 and 12, compressor inlet air temperature and compressor inlet air pressure, are constant values and therefore have been removed. Then, features 6 and 7, starboard propeller torque and port propeller torque, show identical correlation coefficients and indeed they have identical values, so one of them, feature 7, has been removed.

TABLE II. CORRELATION COEFFICIENTS.

Feature ID Number	Compressor Correlation	Turbine Correlation
1	1.54E-18	-2.88E-18
2	-6.20E-19	9.59E-18
3	0.002978258	0.000357638
4	0.001369908	-1.78E-05
5	-0.018837996	0.010000108
6	0.0007535	0.000104423
7	0.0007535	0.000104423
8	-0.03962512	-0.038463904
9	-	-
10	-0.047176568	-0.01685505
11	0.008167586	-0.002718186
12	-	-
13	0.008327871	-0.018303363
14	0.035285243	0.011794227
15	-0.032036625	-0.01887184
16	-0.013667655	-0.017326752

Based on the calculated correlation coefficients, the input features have been reduced by 5 features. Thus, the dimensionality of the input variables has been consequently reduced, an aspect that has been overlooked by the authors in [10].

We emphasize here the importance of data understanding and pre-processing as the use of irrelevant information can only introduce more complexity to the analyzed problem and increase the computational complexity unnecessarily.

## V. REGULARISATION WITH NOISE INTRODUCTION

As discussed before, the gas turbine data has been generated through simulation, therefore it is purely deterministic. In order to mimic the real-world situation where sensors' noise and uncertainty are unavoidable, we added some noise to the outputs (turbine decay coefficient and

compressor decay coefficient). The noise introduced in the output has been generated using Gaussian distributed values of the order of  $10^{-4}$  as the output are defined in the range of  $10^{-3}$ . The main aim of this step is to represent the disturbances, errors and uncertainties in the sensor's readings in real applications.

Some noise has also been introduced to the training data set and added to the 11-correlation based selected input-features as a regularization mechanism for the neural network learning. Indeed, as extensively discussed in [8], introducing noise during the training phase in the input vector can help in controlling the network mapping complexity as well as reducing the probability of data over-fitting.

Given a random vector noise  $n$  and its probability  $p(n)$ , the error used to determine the weights using the error equation 4 for the limit of an infinite number of data points can be re-written as [8]:

$$E = \frac{1}{2} \sum_k \iint [y_k(x) - t_k]^2 \cdot p(t_k|x) \cdot p(x) dx dt_k \quad (9)$$

Then, by introducing the noise:

$$\tilde{E} = \frac{1}{2} \sum_k \iiint [y_k(x+n) - t_k]^2 \cdot p(t_k|x) \cdot p(x) \cdot p(n) dx dt_k dn \quad (10)$$

The noise is usually chosen to be of zero mean and to be uncorrelated between different inputs, therefore:

$$\begin{cases} \int n_i \cdot p(n) \cdot dn = 0 \\ \int n_i \cdot n_j \cdot p(n) \cdot dn = v \cdot \delta_{ij} \end{cases}$$

Using the Taylor series expansion for the error including the noise and considering the zero mean and the noise variance as defined above, then the error including the noise can be expressed as function of the error without the introduction of the noise in the input vector:

$$\bar{E} = E + v\Omega \quad (11)$$

where:

$$\Omega = \frac{1}{2} \sum_k \sum_i \iint \left[ \left( \frac{\delta y_k}{\delta x_i} \right)^2 + \frac{1}{2} (y_k(x) - t_k) \frac{\delta^2 y_k}{\delta x_i^2} \right] \times p(t_k|x) p(x) dx dt_k \quad (12)$$

This is a regularization term added to the usual error (eq. 9). Given that the noise amplitude is small enough to neglect the Taylor expansion high order terms, the minimization of the error with the noise added in the input is equivalent to the

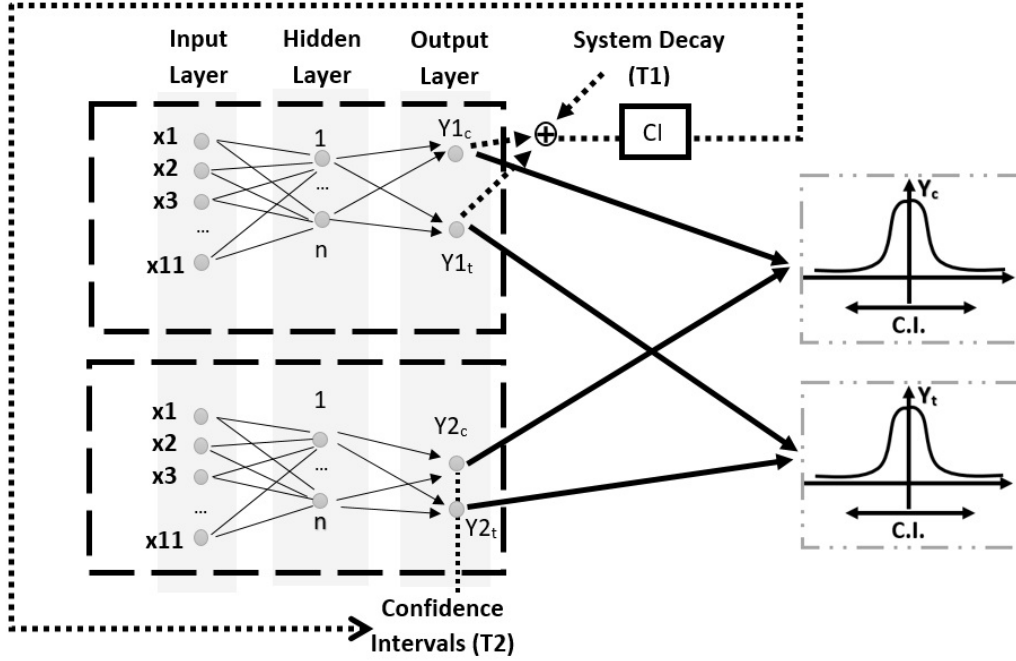


Figure 1. Algorithm structure. Two RBFNN with 11 input features represented by  $(x_1, x_2, \dots, x_n)$  and  $n$  hidden neurons. The top RBF neural network target is represented by the System Decay Coefficients (T1) to be compared with the Compressor Decay Prediction ( $Y_{1c}$ ) and the Turbine Decay Prediction ( $Y_{1t}$ ) to determine the error function. The bottom RBF neural network target is represented by the Confidence Interval (T2) calculated during the training phase of the top RBFNN. This is then compared with the Compressor Confidence Interval prediction ( $Y_{2c}$ ) and the Turbine Confidence Interval prediction ( $Y_{2t}$ ) to calculate the error function.

minimization of the error without the noise terms added to the input plus the regularization term in equation 12. Further details on the definition of regularization functions can be found in [8].

## VI. PREDICTION OF CONFIDENCE INTERVAL

An aspect that is often overlooked is the development of metrics that are suitable for measuring the accuracy of a specific prediction of neural networks. In fact, the development of these metrics allows the introduction of a better level of knowledge of the behavior of the system. When the accuracy of the forecast is not sufficiently precise, alternative decisions can be made in order to avoid worsening the situation if the machine learning algorithm is unable to provide reliable predictions within a certain range of precision. In fact, Neural Networks are not able to automatically provide an assessment of the accuracy of their forecasts. In order to overcome this drawback, numerous methods (Mean-Variance, Delta, Bayesian estimation, and Bootstrap techniques), have been developed, as widely discussed in [12].

In this paper, the Confidence Interval (CI) of the network prediction has been used as a metric of the RBF prediction performance.

The CI is calculated as:

$$CI = \sqrt{(t - y)^2} \quad (13)$$

where:

- $t$  is the target output,
- $y$  is the network predicted output.

In order to be able to predict the confidence interval of the neural network estimates of the system decay, a second RBF network is used to estimate the calculated residual error values between the actual decays and estimated ones after the completion of the training of the RBF that estimates the decays.

Then, the vector of the calculated residual error values has been passed as a target for this second RBF network with the same optimal structure selected for the RBF that predicts the system decays, and with the same 11 features input vector. Once the second RBF network has been trained on the residual error values, the input data for the test phase is passed to the network in order to predict the confidence interval of a prediction given a certain status of the 11 input features. Therefore, the RBF is not only used to predict the equipment status but also the accuracy of the algorithm in performing this prediction.

The proposed RBF structure for estimating the system decays and their confidence intervals is shown in figure 1.

The introduction of the CI allows the careful consideration of the network prediction results focusing the attention on wider CI interval, and therefore less reliable results. In the context of maintenance, this needs to be considered during the decision making related to the best strategy to apply to solve a potential issue in the equipment.

There is not a specific level of confidence defined in the literature and the choice is often dependent on the application. Nevertheless, a common level of prediction accuracy is defined around 95%. This is the level that we aim to achieve in this work, therefore we expect to define a results accuracy level that allows for outliers only for 5% of the cases.

## VII. NUMERICAL SIMULATION: EXPERIMENTS SETUP AND RESULTS

The RBF has been used to predict the system decay and the confidence interval of the prediction. The phases are summarized in Fig.3.

First of all, the RBF network is trained for different number of hidden neurons for both of the noisy input and the clean input. The normalized validation error for the different architectures is then compared to choose the optimal structure. Figure 2 shows the optimal structure (number of hidden neurons) evaluated at different levels of input noise. These optimal structures were calculated such as to achieve a minimum validation error value equal to  $8 \times 10^{-5}$ .

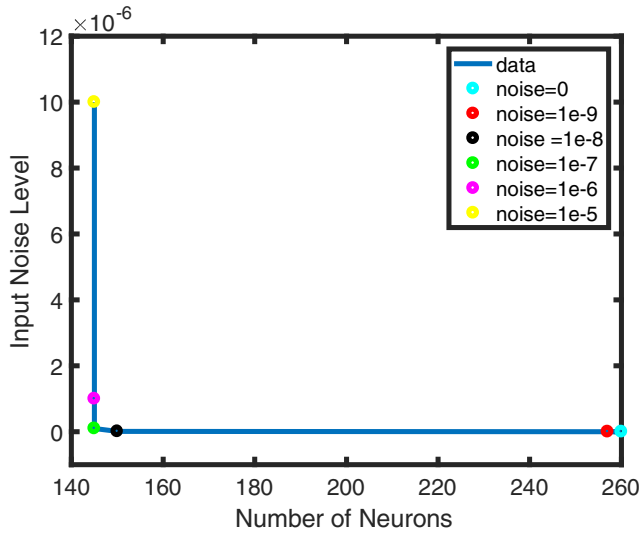


Figure 2. Input noise effect on the complexity of the RBF neural network.

Figure 2 shows that the introduction of a small input noise can help in reducing the complexity of the network as well as limiting the potential over-fitting of the data points. This is consistent with our earlier discussion in Section 5. It can be seen from the figure that if the added noise level is too small, the complexity of the neural network becomes similar to the case where no noise is added to the training input. The complexity of the neural network then decreases with the increase in the noise level. However, although not shown in the figure, when the noise level is increased beyond a certain value (in this case  $1 \times 10^{-5}$ ) the complexity of the neural network increases again. This is expected as the Taylor expansion applies for small noise values only.

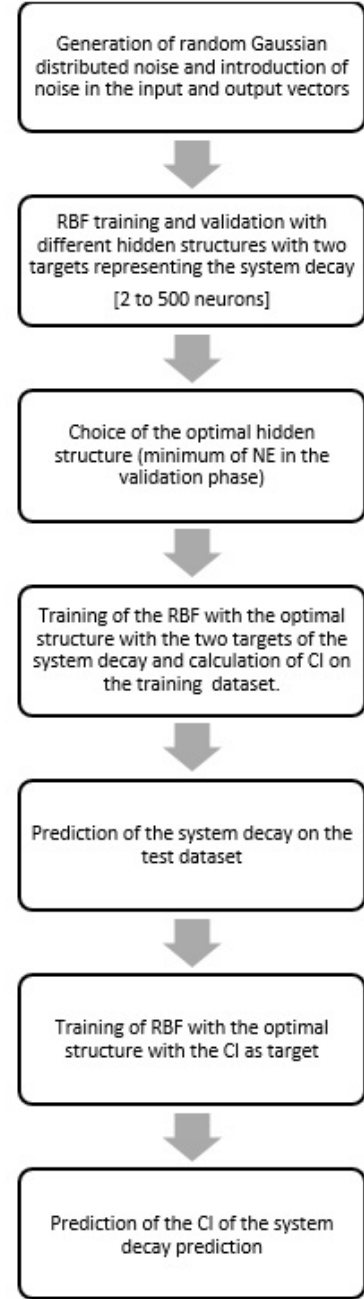


Figure 3. Proposed approach steps with two RBFNN, one dedicated to the prediction of the system performance decay and the second one dedicated to predict the reliability of the prediction of the first RBFNN.

Indeed, for the NE in the case of clean input, the optimal structure can be read from Figure 2 to be 260 hidden neurons, but when a noise level between  $1 \times 10^{-5} - 1 \times 10^{-7}$  is introduced in the input vector, then the optimal structure of the neural network is found to be 145 hidden neurons.

The RBF network is therefore trained using 145 hidden neurons and the results in terms of both Compressor Decay prediction and Turbine decay prediction versus their target vectors are shown in Figure 4 and Figure 5 respectively.

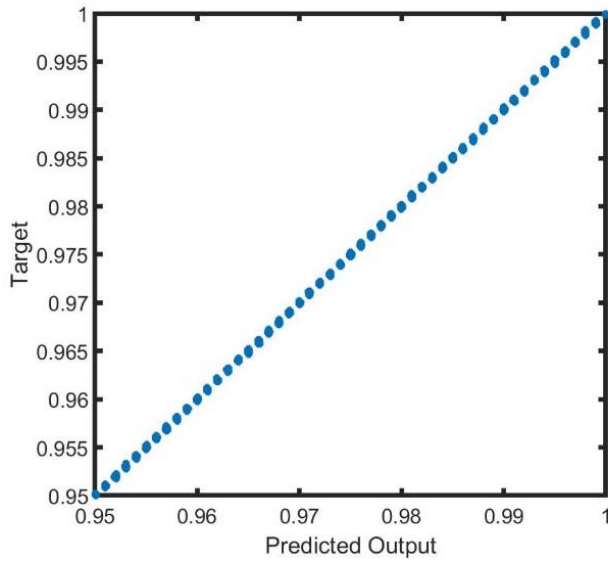


Figure 4. Compressor Prediction Performance. Compressor decay actual output vs RBF predicted output.

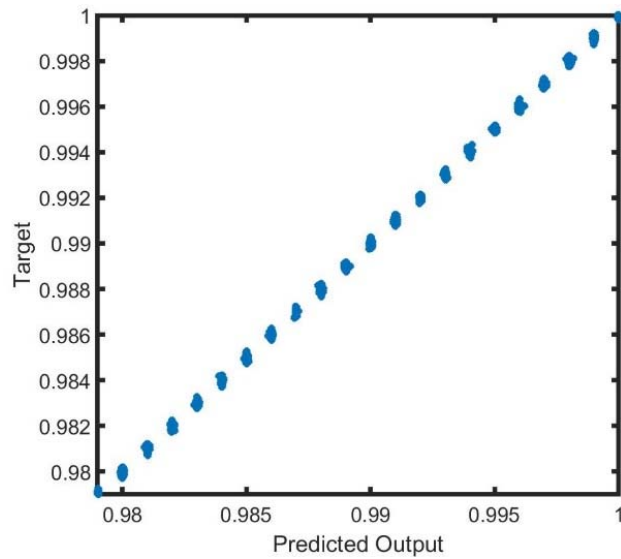


Figure 5. Turbine Prediction Performance. Turbine decay actual output vs RBF predicted output.

The fact that they both lay in straight lines confirms that the algorithm has good capability of predicting the output given a certain status of the selected 11 input features.

Based on our estimates of the compressor and turbine decays from the neural network, we then train another neural network on the residual error between the estimated decays and their actual values. This residual error is then used as the target value for a second RBF network in order to determine the level of accuracy of the system decay prediction on the test dataset.

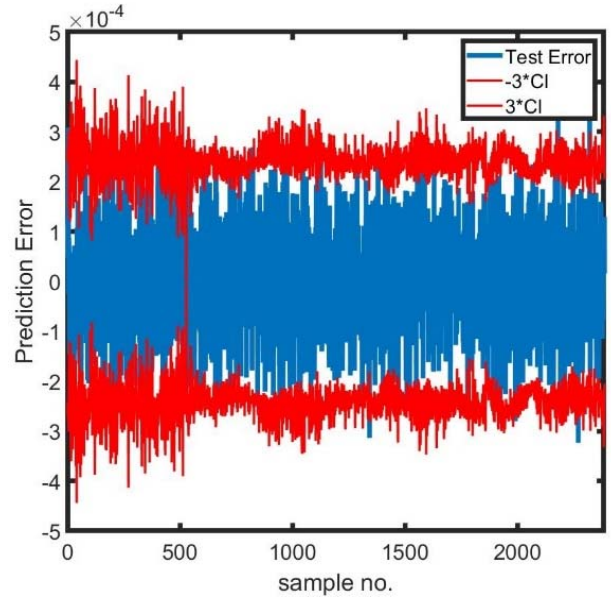


Figure 6. Compressor prediction error and CI. The blue line represents the difference between the estimated turbine decay from the RBF neural network and the actual turbine decay (target vector). The red line represents the chosen level of predicted confidence interval for the same instance.

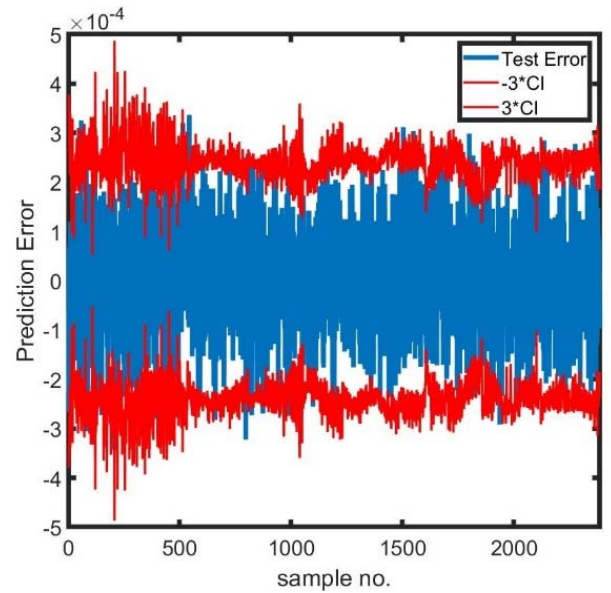


Figure 7. Turbine Prediction error and CI. The blue line represents the difference between the estimated compressor decay from the RBF neural network and the actual compressor decay (target vector). The red line represents the chosen level of predicted confidence interval for the same instance.

As already discussed in the previous paragraphs, the CI in this paper has been chosen in order to obtain a prediction accuracy level  $\geq 95\%$ . For this reason, a  $\pm 3$ -CI level has been chosen. The test error versus the defined level of confidence



for both of the compressor and the turbine are shown in Figure 6 and Figure 7 respectively. In both cases, some outliers can be seen, but they are definitely included in the error limit we set for this application (less than 5%).

The implementation of the proposed architecture where not only the future state of the system but also the accuracy on their predicted values are estimated, introduces a higher level of information for maintenance decision makers. Indeed, the pure prediction of the system status without any indication of the reliability and accuracy of the prediction can lead to false alarms. The incorrect interpretation of these false can lead to poor and potential harmful decisions as well as unneeded stoppages, thus increasing the costs of maintenance operations. With the introduction of the CI, more careful attention can be given to the predictions that present a wider CI and therefore a lower reliability of the prediction. This is fundamental to make sure that only the correct actions are implemented and the production efficiency is maximized.

### VIII. CONCLUSIONS

In the context of Industrial Analytics, the performance prediction of a system represents a highly nonlinear and uncertain problem where the status of several attributes of the system and its components might or might not concur simultaneously to the overall performance decay. An efficient solution to this situation is provided by Neural Networks, which are universal function approximators and therefore well-known to be able to cope with non-linear, uncertain, and big data problems. In particular, a class of Neural Networks called RBF are known to be faster and more stable than other type of Neural Network models such as the Multilayer Perceptron.

The scope of this work is to present an approach to the performance prediction of a system that is able to consider different aspects, starting from the data pre-processing with the application of dimensionality reduction techniques to the use of neural networks as universal approximators. This combined with the estimation of the result reliability through the CI implementation aims to provide a support for the maintenance strategic decision-making process with a reduction in the probability of false alarms.

In this paper, an RBF neural network model for the prediction of a Gas Turbine performance decay has been presented. The importance of understanding and pre-processing the data to select the relevant features only are emphasized. Data pre-processing is shown in this study to help with the reduction of the dimensionality of the input variables, thus reducing computational complexities.

As the gas turbine data is deterministic, we added noise to the decay outputs to reflect real world situations where measured variables are usually corrupted with noise and uncertainty. Furthermore, a noise is added to the input vector to induce a regularization effect during the training phase which in turn helps in reducing the model complexity of the obtained RBF. It is shown that the noise in the input can in fact help in reducing the number of parameters of the trained RBF neural network. Our obtained result on the addition of the input noise to the training input vector is consistent with

the theoretical analysis and provide an evidence for its effectiveness.

The development of a metric to measure the RBF model performance in terms of predicting its accuracy has been discussed as well. It has been implemented through the prediction of the confidence interval. This measure represents the introduction of an improvement in the maintenance function decision-making process as it reduces the probability of false alarms, the number of potentially unneeded stoppages and downtime duration, and enhancing the components and systems availability.

The combination of the confidence interval and the information on the status of the system provide an efficient tool for the development of a CBM with high accuracy in predicting failures in the system. It definitely represents an outstanding improvement in the quality of the decision-making process in the maintenance function supporting the increase of the equipment uptime, reducing the required downtime for the maintenance activities and reducing the related production costs.

An important point to make is that although the developed approach in this paper has been tested only on the Gas Turbine dataset application, the method we developed is definitely transferable to other applications. This is mostly because the studied gas turbine problem is a standard regression problem for which we used and implemented dimensionality reduction techniques and universal function approximators. Furthermore, due to the widespread use of gas turbines for power production, the developed CBM method is relevant to a large number of applications including large machine producers and users.

### REFERENCES

- [1] J. Metternich, J. Böllhoff, S. Seifermann, and S. Beck, "Volume and Mix Flexibility Evaluation of Lean Production Systems," *Procedia CIRP*, vol. 9, pp. 79-84, 2013.
- [2] M. Celen and D. Djurdjanovic, "Operation-dependent maintenance scheduling in flexible manufacturing systems," *CIRP Journal of Manufacturing Science and Technology*, vol. 5, no. 4, pp. 296-308, 2012.
- [3] M. Savsar, "Effects of maintenance policies on the productivity of flexible manufacturing cells," *Omega*, Article vol. 34, pp. 274-282, 1/1/2006 2006.
- [4] X. Y. Xu and Q. S. Hua, "Industrial Big Data Analysis in Smart Factory: Current Status and Research Strategies," (in English), *Ieee Access*, Article vol. 5, pp. 17543-17551, 2017.
- [5] P. Ashok, B. James, and G. Subramaniam, "Condition based maintenance: a survey," *Journal of Quality in Maintenance Engineering*, no. 4, p. 384, 2012.
- [6] W. Zhang, M. P. Jia, L. Zhu, and X. A. Yan. (2017) Comprehensive Overview on Computational Intelligence Techniques for Machinery Condition Monitoring and Fault Diagnosis. [Periodical]. 782. Available: <http://search.ebscohost.com/login.aspx?direct=true&db=edsbl&AN=RN384546443&site=eds-live&authtype=ip,shib&custid=s9815128>
- [7] K. Verbert, B. De Schutter, and R. Babuška, "Timely condition-based maintenance planning for multi-component systems," *Reliability Engineering & System Safety*, vol. 159, pp. 310-321, 2017.



- [8] C. M. Bishop, Neural networks for pattern recognition. Oxford University Press, 1995., 1995.
- [9] A. Markopoulos, S. Georgiopoulos, and D. Manolakis, "On the use of back propagation and radial basis function neural networks in surface roughness prediction," *Journal of Industrial Engineering International*, vol. 12, no. 3, p. 389, 09// 2016.
- [10] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, and M. Figari, "Machine learning approaches for improving condition-based maintenance of naval propulsion plants," (in English), *Proceedings of the Institution of Mechanical Engineers Part M-Journal of Engineering for the Maritime Environment*, Article vol. 230, no. 1, pp. 136-153, Feb 2016.
- [11] C. David Camilo, C. Juan Carlos, and L. Agapito, "How to Address the Data Quality Issues in Regression Models: A Guided Process for Data Cleaning," *Symmetry*, Vol 10, Iss 4, p 99 (2018), article no. 4, p. 99, 2018.
- [12] "Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances," *IEEE Transactions on Neural Networks, Neural Networks, IEEE Transactions on, IEEE Trans. Neural Netw., Periodical no. 9*, p. 1341, 2011.